

## ระบด

0.4 second, 512 MB

เมืองแห่งหนึ่งมีประชากร  $N$  คน ( $1 \leq N \leq 100,000$ ) (มีหมายเลข 0 ถึง  $N-1$ ) หนึ่งในประชากรมีเชื้อโรคระบาดอยู่ ตอนนี้ทุกคนถูกสั่งให้รักษาระยะห่างทางกายภาพ จึงจะไม่มีการติดต่อเกิดขึ้นอีก (ในโลกสมมติใบนี้)

อย่างไรก็ตาม ถ้าต้องการจะลดมาตรการนี้ ทางเมืองต้องหาให้ได้ว่าใครคือผู้ติดเชื้อคนนั้น มีการรวบรวมกำลังอาสาสมัครจำนวน  $K$  คนจากเมืองอื่น ๆ (มีหมายเลข 0 ถึง  $K-1$ ) มาเพื่อรับความเสี่ยงในการตามหาผู้ติดเชื้อ กล่าวคือ เราสามารถบอกให้อาสาสมัครแต่ละคนเดินทางเข้าไปใกล้ประชากรคนใดก็ได้ จากนั้นจากข้อมูลการติดเชื้อของอาสาสมัคร เราจะพยายามหาว่าประชากรคนใดคือคนที่ติดเชื้อ

เราทราบพฤติกรรมของโรคดังนี้ ถ้าอาสาสมัครเข้าไปใกล้ผู้ติดเชื้อในวันที่  $i$  เมื่อสิ้นวันที่  $i+30$  อาสาสมัครจะตรวจพบโรค (เมื่อติดโรคแล้วจะไปพบคนอื่นไม่ได้อีก) เป้าหมายของเมืองคือต้องการจะหาให้ได้ว่าใครคือผู้ติดเชื้อหลังจากเวลาผ่านไปครบ 34 วัน (เมืองต้องการจะยกเลิกการ lock down ในวันที่ 35)

## งานของคุณ

ข้อนี้เป็นโจทย์ interactive ใน 2 subtask แรก (subtask 1 และ 2) อาสาสมัครจะสามารถพบกับคนที่คนอื่นได้ในแต่ละวัน ฟังก์ชันที่คุณเขียนจะต้องเรียก API รวมไม่เกิน 34 รอบ ในแต่ละรอบ (แทนแต่ละวัน) คุณจะเขียนโปรแกรมระบุให้อาสาสมัครแต่ละคนไปพบกับประชากรคนใดบ้าง และจะได้รับผลการตรวจ (เมื่อสิ้นวัน) ของอาสาสมัครทุกคน คุณต้องระบุประชากรที่ติดเชื้อให้ได้ สำหรับ subtask 1,  $K = 20$  และ subtask 2,  $K = 15$

ใน subtask ถัด ๆ ไปจะมีข้อจำกัดว่าในแต่ละวัน ประชากรคนหนึ่ง ๆ จะต้องเจออาสาสมัครไม่เกิน  $L$  คน (เพื่อไม่ให้ประชาชนแตกตื่น) ใน subtask 3,  $L = 4$ , และ  $K = 15$

สำหรับ subtask 4, 5, และ 6 เราจะมีอาสาสมัครจำนวนไม่จำกัด (ไม่เกิน 100,000 คน) แต่มีค่า  $L = 3, 2$ , และ 1 ตามลำดับ คะแนนที่คุณได้จะคิดจากจำนวนอาสาสมัครที่คุณใช้ในการติดตามโรค (ยิ่งใช้อาสาสมัครมากยิ่งได้คะแนนน้อย) คะแนนของ subtask เหล่านี้จะประกาศเมื่อเสร็จสิ้นการแข่งขัน

## การติดต่อกับไลบรารี

คุณจะได้รับ include file `pandelib.h` (ให้ include ในโปรแกรมของคุณ) และ `pandelib.cpp` (ให้คอมไพล์ร่วมกับโปรแกรมของคุณ) จะมีฟังก์ชันดังนี้

- `pandemic_init(int& N, int& K, int& L)` – อ่านค่าเริ่มต้น สามารถตรวจสอบว่าอยู่ subtask ใดได้จากค่า  $K$  และ  $L$
- `pandemic_assign(assignment_t assignments[], bool results[])` – ใช้เรียกแต่ละวัน เริ่มจากวันที่ 1 โดยระบุอาร์เรย์ของ `assignment_t` ที่แต่ละ element เป็น vector ของจำนวนเต็ม (`vector<int>`) ส่วนอาร์เรย์ `results` เป็นอาร์เรย์ที่จะคืนค่ามาจะเป็นสถานะการติดโรคเมื่อสิ้นวันนั้น (เรียก `assignment` วันที่  $i$  จะเห็นผลสิ้นวันที่  $i$ ) อาสาสมัครที่ติดโรคแล้วจะติดตลอดไป เรียกได้ไม่เกิน 34 ครั้ง

- `pandemic_answer(int ans)` – ตอบคำถามว่าใครติดโรค สามารถเรียกได้ก่อนจะครบ 34 วัน เรียกแล้วโปรแกรมจะจบการทำงานทันที

### ตัวอย่างการทำงาน 1

สมมติว่า  $N = 5$ ,  $K = 5$ ,  $L = K$  (ไม่จำกัดจำนวนอาสาสมัครที่แต่ละคนเจอในแต่ละวัน) และคนที่มิเชื้อคือคนหมายเลข 3

call	output	คำอธิบาย
<code>pandemic_init(N,K,L)</code>	$N=5, K=5, L=5$	
<code>pandemic_assign([[0,1],[1,2],[2,3],[3,4],[4,0]], results)</code>	$results=[F,F,F,F,F]$	วันที่ 1 (results สิ้นวันที่ 1)
<code>pandemic_assign([[0,1,2,3,4],[],[[[]], results)</code>	$results=[F,F,F,F,F]$	วันที่ 2 (results สิ้นวันที่ 2)
<code>pandemic_assign([[],[],[],[],[]], results)</code>	$results=[F,F,F,F,F]$	วันที่ 3 (results สิ้นวันที่ 3)
...	...	...
<code>pandemic_assign([[],[],[],[],[]], results)</code>	$results=[F,F,F,F,F]$	วันที่ 30 (res สิ้นวันที่ 30)
<code>pandemic_assign([[],[],[],[],[]], results)</code>	$results=[F,F,T,T,F]$	วันที่ 31 อาสาสมัครคนที่ 2 และ 3 ได้พบกับคนที่ 3 ในวันที่ 1
<code>pandemic_assign([[],[],[],[],[]], results)</code>	$results=[T,F,T,T,F]$	วันที่ 32 อาสาสมัครคนที่ 0 พบกับคนที่ 3 ในวันที่ 2
<code>pandemic_answer(3)</code>		ตอบถูกต้อง

ในการตรวจ ถ้าโปรแกรมถูกต้อง ใน test cast ใน subtask จะระบุจำนวนอาสาสมัครที่ใช้ ถ้าไม่ใช่จำนวนจะเป็น - คือตอบผิด, c คือมีบางคนพบกับอาสาสมัครเกิน L คนในหนึ่งวัน และ i คือมีการเรียก assign มากกว่า 34 ครั้ง

### ปัญหาย่อย

- ปัญหาย่อย 1 (10%):  $K = 20, L = 20$
- ปัญหาย่อย 2 (30%):  $K = 15, L = 15$
- ปัญหาย่อย 3 (30%):  $K = 15, L = 4$
- ปัญหาย่อย 4 (10%):  $K = 100,000, L = 3$  (ให้คะแนนจากจำนวนอาสาสมัครที่ใช้)
- ปัญหาย่อย 5 (10%):  $K = 100,000, L = 2$  (ให้คะแนนจากจำนวนอาสาสมัครที่ใช้)
- ปัญหาย่อย 6 (10%):  $K = 100,000, L = 1$  (ให้คะแนนจากจำนวนอาสาสมัครที่ใช้)

ในทุกปัญหาย่อย ไม่จำเป็นที่ N จะต้องเท่ากับ 100,000 เสมอไป

(มีต่อหน้าถัดไป)

### การให้คะแนน

สำหรับปัญหาย่อย 4, 5 และ 6 ถ้าคุณใช้อาสาสมัครมากที่สุด  $Q$  คน และมีโปรแกรมของผู้แข่งขันหรือของกรรมการที่ใช้อาสาสมัครมากที่สุดน้อยที่สุดคือ  $P$  คน คุณจะได้คะแนน  $(P/Q)^2$  ของคะแนนเต็มในปัญหาย่อยนั้น

### การใช้งานไลบรารีสำหรับทดสอบ

สามารถดาวน์โหลดได้ที่ (ดูในประกาศ) โค้ดเหล่านี้คุณสามารถแก้ไขได้เพื่อช่วยในการตรวจสอบโปรแกรม  
ให้ป้อนข้อมูลเข้าในรูปแบบนี้

N K L

R

โดยที่ R คือหมายเลขคนที่ติดเชื้อ

### ตัวอย่าง

Input
5 5 5 3